Name: Sam Brasher

Class: CS 340-T6642

Assignment: Project Two Readme

# Required Functionality

The program is required to display a database of animal data in clinics around the Austin area, sort through that data and allow the user to filter by which animals might be best for various search and rescue tasks – disaster or individual tracking, mountain or wilderness, and water rescue. That data should be displayed in a table containing all stored information on the animal, a map showing the location of the animal, and a pie chart showing breeds available after selection. Finally, the user should be able to reset the displays and return to showing the entire dataset. Screenshots showing this functionality can be found at the end of this document.

# Tools Used

* MongoDB provides a non-relational database for storing various types of data related to animals. There is also a framework, PyMongo, that is a well-tested and recommended way to work with MongoDB from Python.
  + https://pymongo.readthedocs.io/en/stable/
* Dash is a low-code way to build user interfaces in data applications. It allows for use of HTML along with various interactive charts, graphs, and tables for programmers to design a visually appealing app simply and quickly.
  + <https://dash.plotly.com/introduction>
* Pandas is a tool for data analysis in Python. Combined with Dash, it allows us to create rich and informative diagrams that connect the user to relevant information.
  + https://pandas.pydata.org/
* Jupyter Notebook is the IDE used for developing this application.
  + https://jupyter.org/

# Project Creation Timeline

1. First, we properly set up a Mongo database and load the animal center data into it.
2. Next, we created a CRUD interface class in Python that, following a MVC model, allows a controller to affect our model by creating new entries, reading and updating existing entries, and deleting entries.
3. Next, we focused on creating a datatable in Dash, using a dataframe from Pandas, to display our data to the user.
4. We then used a leaflet map from Dash, taking input from our new datatable, that allows the user to see where exactly the animal is located. At that time, the map only displayed the first animal in whatever page the user was on.
5. Next, we enabled filtering options for the user that affected the data displayed in the datatable i.e., showing different breeds of dogs for different tasks. At the same time, we allowed the user to select a specific entry in the datatable to give interactivity to the geolocation map.
6. Finally, we created a pie chart that gave overall trends on what data was loaded into the datatable.

# Challenges

A major challenge was the use of JupyterDash vs Dash in our system. After attempting to use JupyterDash without success for several days, a switch was made to Dash that solved several problems.

# Screenshots

A picture containing graphical user interface

Description automatically generated

Figure 1: Showing Disaster or Individual Tracking

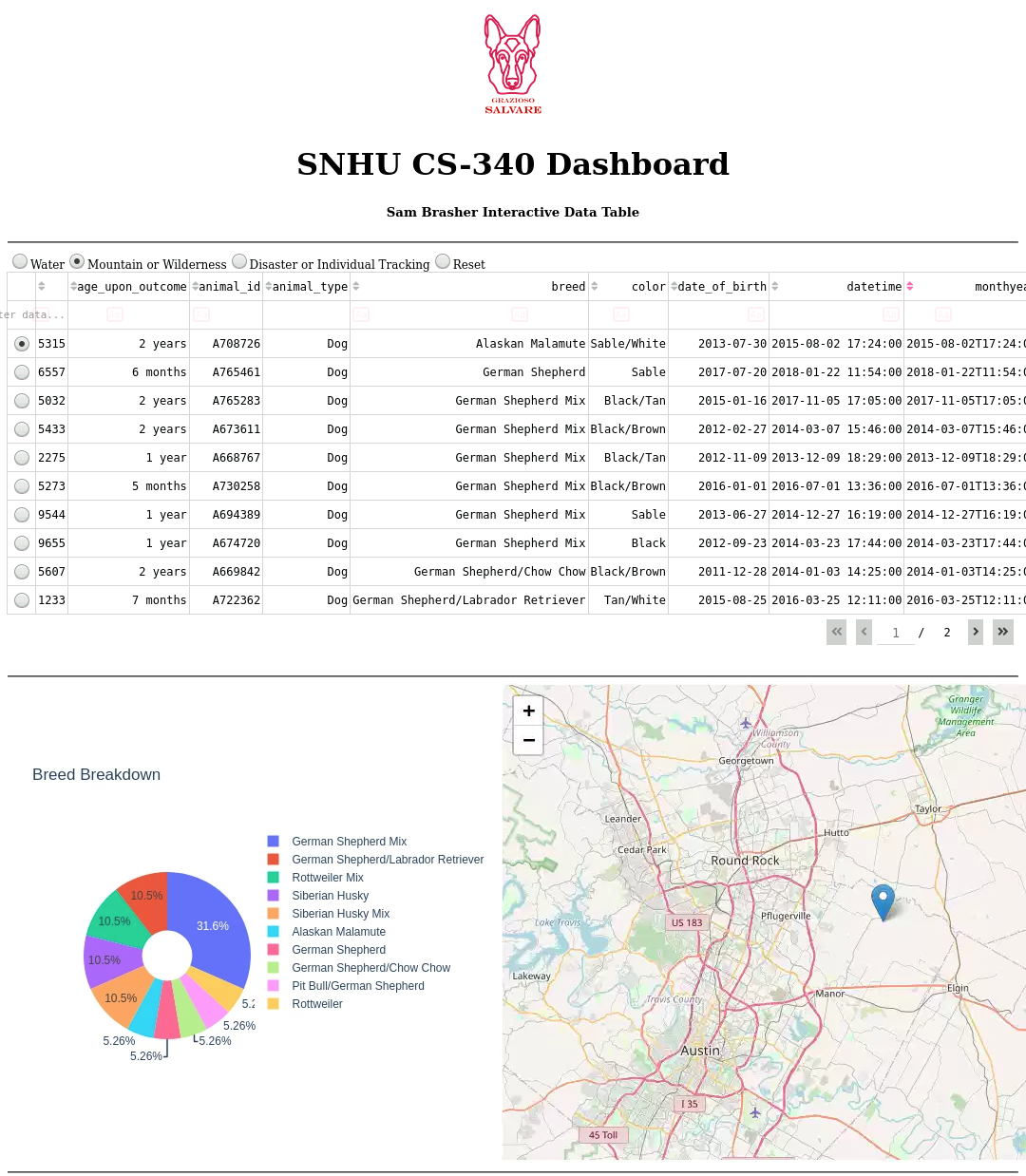


Figure 2: Showing Mountain or Wilderness Rescue

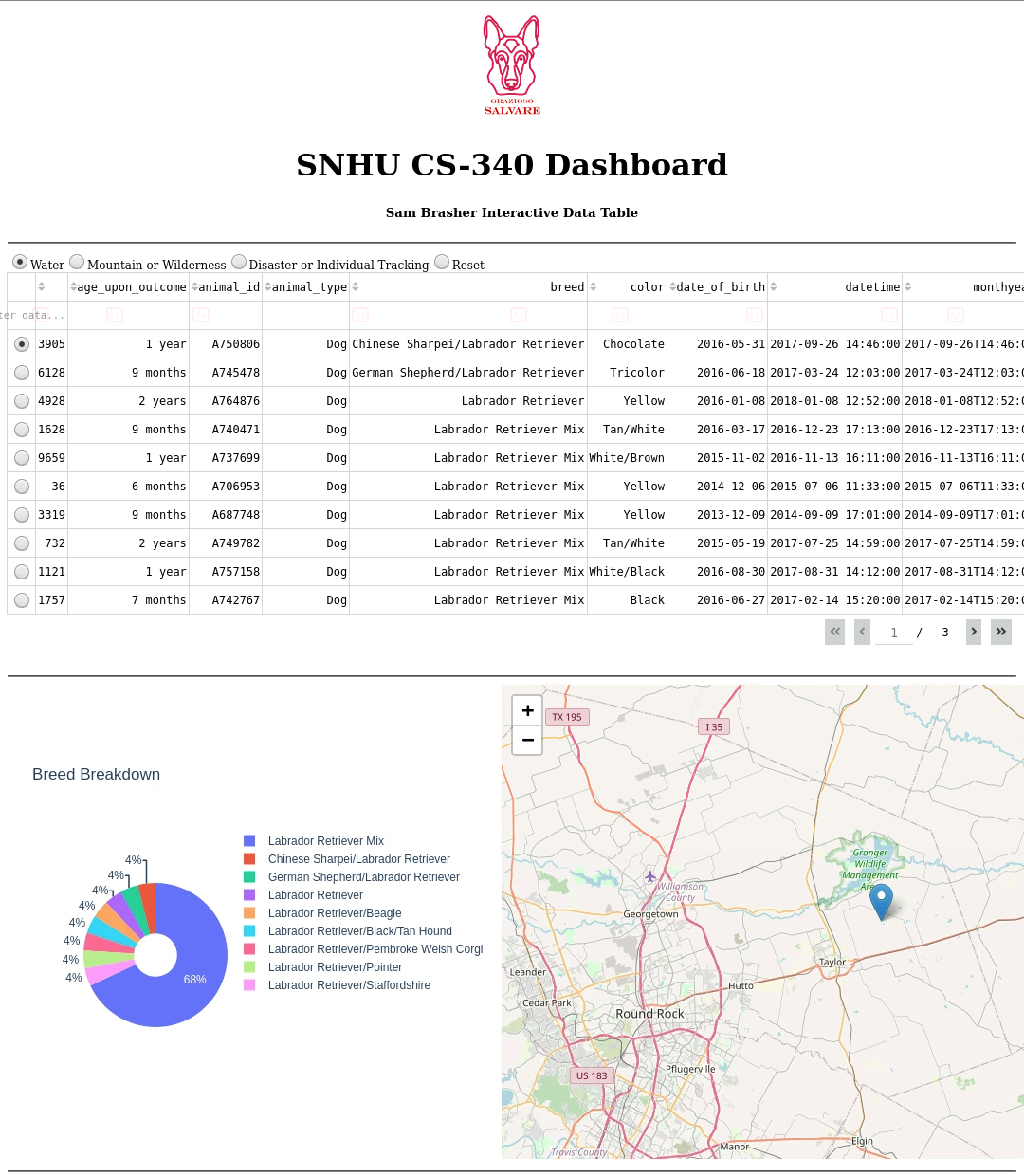


Figure 3: Showing Water Rescue

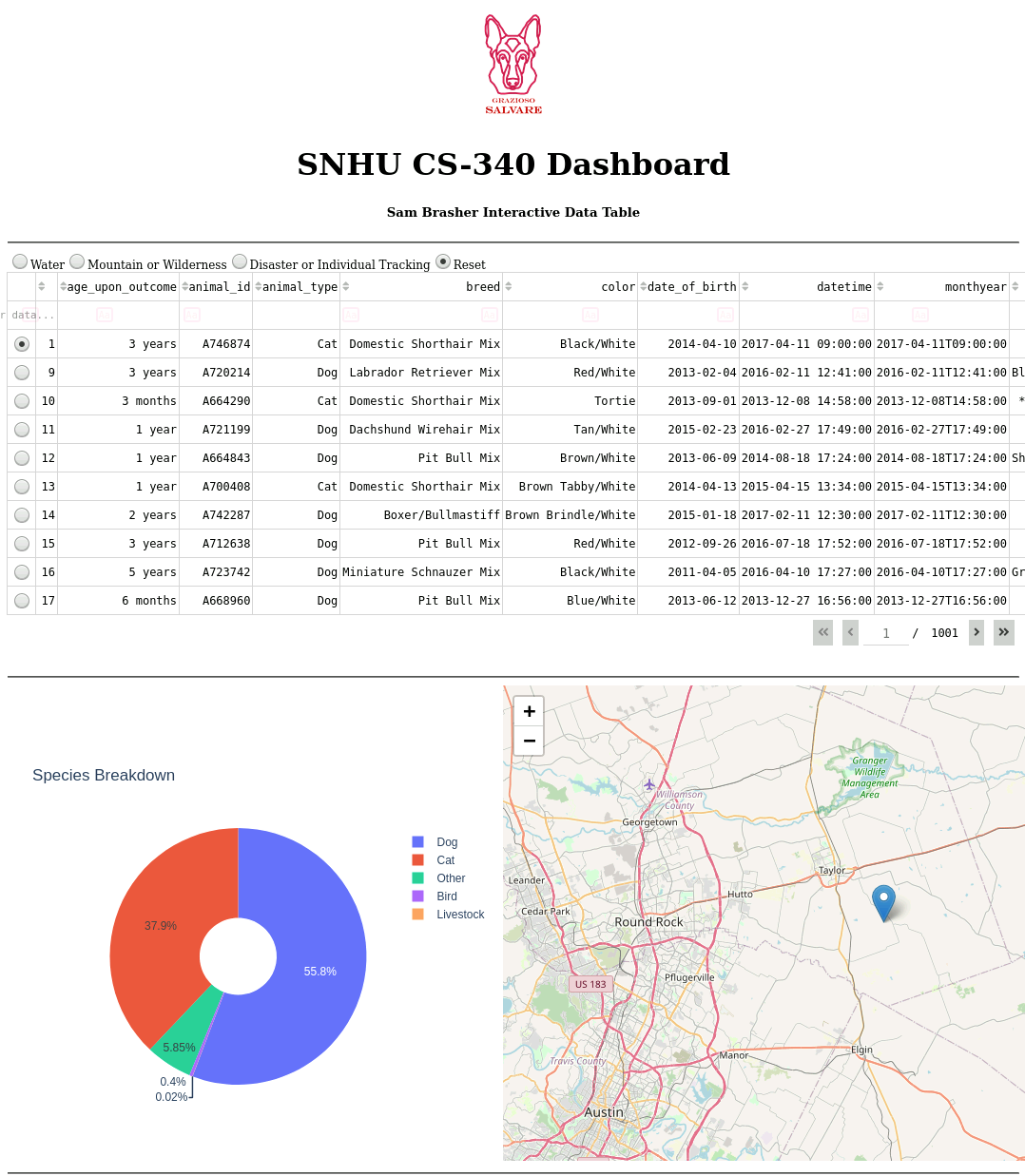


Figure 4: Showing "Reset" Functionality